

REMARKS

[0002] Applicant respectfully requests reconsideration and allowance of all of the claims of the application. Claims 1-11 and 13-23 are presently pending. Claims amended herein are 2-3 and 20-21. No claims are withdrawn, cancelled or added herein.

Claim Amendments

[0003] Without conceding the propriety of the rejections herein and in the interest of expediting prosecution, Applicant amends claims 2-3 and 20-21 herein. Applicant amends claims to clarify claimed features. Such amendments are made to expedite prosecution and more quickly identify allowable subject matter. Such amendments are merely intended to clarify the claimed features, and should not be construed as further limiting the claimed invention in response to the cited references.

Formal Matters

[0004] This section addresses any formal matters (e.g., objections) raised by the Examiner.

Specification

[0005] The Examiner objects to the paragraph beginning on page 16, line 17 of the specification for missing referenced examples. Herein, Applicant amends this paragraph, as shown above, to correct the informalities noted by the Examiner.

Claims

[0006] The Examiner objects to claims 2, 3 and 20 for insufficient antecedent basis for the limitation "the executing act." Herein, Applicant amends these claims, as shown above, to address the objection made by the Examiner, and to expedite prosecution.

Substantive Matters

Claim Rejections under § 103

[0007] The Examiner rejects claims 1-11 and 13-23 under § 103. For the reasons set forth below, the Examiner has not made a prima facie case showing that the rejected claims are obvious.

[0008] Accordingly, Applicant respectfully requests that the § 103 rejections be withdrawn and the case be passed along to issuance.

[0009] The Examiner's rejections are based upon the following references alone or in combination:

- **Murray:** *Murray, et al.*, US Patent Application Publication No. 2006/0235968 (Published October 19, 2006);
- **Young:** *Young*, US Patent No. 6,782,531 (issued August 24, 2004);
and
- **Kriens:** *Kriens*, US Patent No. 5,864,862 (issued January 26, 1999).

Overview of the Application

[0010] The Application describes various capabilities for resolving strings within a command string. The present mechanism operates within an interactive operating environment by receiving a plurality of strings. For any string that is partially resolved, the mechanism initiates analysis for completely resolving the string. The mechanisms support wildcarding, property sets, relations, conversions, property paths, extended types, data type coercing and the like.

Cited References

[0011] The Examiner cites Murray as the primary reference in the obviousness-based rejections. The Examiner cites Young and Kriens as secondary references in the obviousness-based rejections.

Murray

[0012] Murray describes a method of interacting with a managed data network entity. The method includes a sequence of steps. A change in the operational state of the managed data network entity is detected. A command line interface (CLI) dictionary entry is retrieved from a CLI dictionary associated with the data network entity. Based on the retrieved CLI dictionary entry, CLI commands are extracted therefrom to configure the managed data network entity to reflect the detected change in the operational state.

[0013] A CLI command sequence is built from the extracted CLI commands. Each CLI command in the command sequence is sent to the managed data network entity for execution. CLI command responses are monitored. Based on a successful execution of CLI commands sent, subsequent CLI commands in the CLI command sequence are sent for execution.

Young

[0014] Young describes a technology for ordering data processing by multiple "plug-in" processing modules controlled by an execution management framework. The framework includes an order determining mechanism that controls the plug-in processing modules to execute in series, or in parallel, to speed processing by the plug-ins while accommodating computational dependencies.

[0015] In one embodiment, the order determining mechanism can include a dependency counter associated with each plug-in processing module for determining an operational sequence position of the associated plug-in processing module, and means for conditioning the count value (e.g., decrementing or incrementing the count value) in response to operation of another plug-in processing module on whose output the associated plug-in processing module depends, so that the associated plug-in processing module will commence operation in the proper order when the count value reaches a predetermined value.

Kriens

[0016] Kriens describes a technology for creating reusable components in an object-oriented programming environment. An object called the Data Object is used to represent different information types in a system in a singular way in an object-oriented programming environment. The Data Object stores the value

of the information and a reference to an object that handles that specific type of information. All requests on the Data Object are forwarded to this handler.

[0017] The interface allows programmers to query the structure of the information, place constraints on values and names of the parts. This makes it possible to write code that performs a service that adapts itself to whatever Data Object it gets. The use of a Data Object significantly minimizes the coupling between modules and make reusable code easier to develop because each object class in the object-oriented system is not coupled only to the Data Object class.

Obviousness Rejections

Lack of *Prima Facie* Case of Obviousness (MPEP § 2142)

[0018] Applicant disagrees with the Examiner's obviousness rejections. Arguments presented herein point to various aspects of the record to demonstrate that all of the criteria set forth for making a prima facie case have not been met.

Based upon Murray, Young and Kriens

[0019] The Examiner rejects claims 1-11 and 13-23 under 35 U.S.C. § 103(a) as being unpatentable over Murray, in view of Young and further in view of Kriens. Applicant respectfully traverses the rejection of these claims and asks the Examiner to withdraw the rejection of these claims.

Independent Claim 1

[0020] Applicant submits that the combination of Murray, Young and Kriens does not teach or suggest at least the following features as recited in this claim (with emphasis added):

- "parsing a sequence of ***object-based commands*** into **individual *object-based commands***"
- "executing **each execution element** associated with each individual ***object-based command*** to produce output objects, wherein the

execution of each execution element is execution dependent upon an execution-supporting operating environment”

[0021] The Examiner indicates (Action, p. 3-4) the following with regard to this claim:

As to claims 1, and 19, Murrav et al. discloses a system that extends data types available to an operating environment, the system comprising:
a processor (See Murray et al. page 3, paragraph 0024); and
a memory, the memory being allocated for a plurality of computer-executable instructions which are loaded into the memory (See Murrav et al. page 3, paragraph 0024) for execution by the processor, the computer-executable instructions comprising:

parsing a sequence of object-based commands into individual object-based commands (See Murray et al. page 7, paragraph 0090);

associating each individual object-based command with at least one execution element (See Murray et al. page 6, paragraph 0085).

[0022] Applicant respectfully submits that Murray fails to teach or suggest “parsing a sequence of ***object-based commands*** into ***individual object-based commands***.” As discussed in the Specification, one of the advantages of “object-based” commands is that the executable cmdlets associated with the individual object-based commands require less code than commands in prior administrative environments. See, for example, Specification at page 46 lines 15-17. Cmdlets are typically smaller in size than their counterpart commands because the cmdlets can utilize common functions provided by the administrative tool framework, such as parsing, data validation, error reporting, and the like. See, for example, Specification at page 8 lines 17-25. Because such common functions can be implemented once and tested once, the use of cmdlets throughout the administrative tool framework allows the incremental

development and test costs associated with application-specific functions to be quite low compared to traditional environments.

[0023] Murray at most describes building CLI command sequences 544 from multiple CLI actions 266. See, for example, Murray at paragraph [0090]. The CLI actions 266 are mapped to basic actions 262 which includes: create, update, read, delete actions, etc. See, for example, Murray at paragraphs [0065]-[0067].

[0024] Nowhere does Murray describe the “actions” or “command sequences” as being ***object-based***. In fact, Murray explicitly **teaches against** using object-oriented programming by stating: “heavy reliance on object oriented programming to achieve compact code intending to reduce the size of software applications and perhaps development time, suffers from deeply nested function calls which creates a processing overhead leading to inefficient code.” See, for example, Murray at paragraph [0013]. It further states “[d]eep nesting of function calls obscures the implementation paradigms used; thereby negatively impacting code debugging, code maintenance, and further development thereof.” Id. Therefore, Murray fails to teach or suggest “parsing a sequence of ***object-based commands*** into ***individual object-based commands***.”

[0025] In addition, the Examiner admits that Murray fails to teach “executing each execution element ***associated with each individual object-based command*** to produce output objects, wherein the execution of each execution element is execution dependent upon an execution-supporting

operating environment.” The Examiner relied on Young to compensate for the defects of Murray. Applicant respectfully disagrees.

[0026] Young at most describes a transaction processor pipeline 130 made up of a number of pipeline stages 202, 204, 206 under the control of a pipeline controller 210. See, for example, Young at Col. 7 lines 15-61. Each of the stages 202, 204, 206 coordinates the execution of a number of processing modules (“plug-ins”) 220, supported by an execution management framework 225. The plug-ins are modular, computer-executable objects each containing a process for performing a single function. Therefore, each plug-in is associated with a **pipeline stage**, which is merely a step that transforms metered data contents into processed usage data, and not an **individual object-based command** derived by parsing a sequence of object-based commands. See, for example, Young at Col. 5 lines 9-12.

[0027] Kriens fails to compensate for the defects of Murray and Young. Therefore, as shown above, the combination of Murray, Young and Kriens does not teach or suggest all of the elements and features of this claim. Accordingly, Applicant asks the Examiner to withdraw the rejection of this claim.

Independent Claims 14 and 19

[0028] Claims 14 and 19 recite features similar to those of claim 1. Accordingly, at least for the same reasons described above with regard to claim 1, claims 14 and 19 are patentable over Murray, Young and Kriens.

Dependent Claims 2-11, 13, 15-18 and 20-23

[0029] These claims ultimately depend upon independent claims 1, 14 and 19. As discussed above, claims 1, 14 and 19 are allowable. It is axiomatic that any dependent claim which depends from an allowable base claim is also allowable. Additionally, some or all of these claims may also be allowable for additional independent reasons.

Conclusion

[0030] All pending claims are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the application. If any issues remain that prevent issuance of this application, the **Examiner is urged to contact me before issuing a subsequent Action.** Please call or email me at your convenience.

Respectfully Submitted,

Lee & Hayes, PLLC
Representatives for Applicant

/kaseychristie40559/

Dated: 2/11/2009

Kasey C. Christie (kasey@leehayes.com; 509-944-4732)

Registration No. 40559

Customer No. **22801**

Telephone: (509) 324-9256

Facsimile: (509) 323-8979

www.leehayes.com